

# ALE 기반 RFID 미들웨어 시스템 설계

## A Design of ALE-compliant RFID Middleware System

홍연미, 조윤상\*, 변지웅, 노영식, 박상열\*, 오상현\*\*, 변영철  
 제주대학교, 자바정보기술(주)\*, 연세대학교\*\*

Hong Yeon-Mi, Cho Yoon-Sang\*, Byun Ji-Woong,  
 No Young-Sik, Park Sang-Yeol\*, Sang-Hyun Oh\*\*,  
 Byun Yung-Cheol  
 Cheju National University, Java Information  
 Technology, Ltd.\*, Yonsei University\*\*

### 요약

유비쿼터스 컴퓨팅을 실현하기 위한 핵심 기술인 자동식별, 센서 네트워크, 홈 네트워크, 텔레매틱스 등 다양한 분야에서 연구가 진행되고 있다. 특히 RFID 기술을 이용한 기존 솔루션과의 통합 및 데이터 수집, 제어, 관리 등을 지원하는 RFID 미들웨어의 중요성에 대한 인식이 확산되고 있다. 한편 EPCglobal에서는 기존의 Savant를 대체하는 개념으로서 RFID 하드웨어와 응용 사이에 존재하는 ALE(Application Level Events) 표준 스펙을 정의하였다. 이는 레이어의 내부 구조 및 구현 기술 등은 명시하지 않고 오직 외부 인터페이스만을 정의함으로써 향후 상호 운용성 및 표준 적합성 검증을 용이하게 한다. 본 논문에서는 이러한 ALE 스펙에 기반하여 RFID 데이터를 효율적으로 처리할 수 있는 RFID 미들웨어를 제안한다.

### Abstract

Nowadays, to realize ubiquitous computing environment, many research activities have been going on within various kinds of research domains including automatic identification, sensor network, home network, telematics and so on. Especially, RFID middle-ware that supports the aggregation of RFID tag data, control and management, and the integration with legacy systems has recently gained a lot of attention. Meanwhile, EPCglobal defined an ALE(Application Level Events) standard specification, which exists between RFID readers and applications, and substitutes the previous systems called Savant. In the specification internal structures and implementation technologies of ALE are not mentioned and only external interfaces are defined. This approach eases the verification of standard compliance and inter-operability of the layer. In this paper, we present the design of ALE-compliant RFID middle-ware systems that process RFID tag data efficiently.

## I. 서론

유비쿼터스 컴퓨팅을 실현하기 위하여 자동식별, 홈 네트워크, 센서 네트워크, 텔레매틱스 등 다양한 분야에서 연구가 활발히 이루어지고 있다. 특히 자동식별 분야에 있어서 사람 및 사물의 신원 정보를 제공할 수 있는 RFID가 주목받고 있다.

무선 주파수 인식 기술인 RFID(Radio Frequency Identification)는 20세기 중반에 개발되어 1990년대 말에 재고 관리 및 공급 체인 관리 등에서 사용됨으로써 주목을 받은 기술이다. RFID 시스템은 기존 바코드 시스템의 한계성 및 인식 방식에 있어서의 문제점을 극복하였다. 과거에는 시스템이 사물의 실체를 인식할 수 없었지만 사물에 RFID 태그를 부착함으로써 사물에 대한 정보를 자동으로 인식하고 네트워크에 연결하여 실시간으로 정보를 관리함으로써 편리한 서비스를 제공

할 수 있게 되었다.

초기의 RFID 기술 연구 및 시장 형성은 주로 사물에 부착하기 위한 태그와 이를 무선을 통해 자동으로 인식하기 위한 칩, 리더 등의 하드웨어 중심으로 발전 되어왔다. 그러나 최근 기존 시스템과 RFID 시스템 간의 통합을 지원하기 위해 태그 데이터의 수집, 정제 및 관리 등을 수행하는 미들웨어에 대한 관심과 수요가 증가하고 있다. 하지만 기존 RFID 미들웨어의 모델은 Auto-ID Lab에서 제안한 Savant 모델을 기반 한 데이터 수집, 가공처리에 국한된 RFID 미들웨어로서 기본적인 기능만 갖춰진 형태로 향후 리더기 혹은 응용 서버 등 다양한 외부 비즈니스 로직과의 상호 운영성이 많이 결여되는 문제점이 대두되었다.

최근 EPCglobal에서 진행하고 있는 EPC 네트워크 구성 요

소 중 RFID 미들웨어와 응용 사이에 존재하는 ALE가 기존 Savant의 개념을 대체하였는데, ALE 스펙에서는 미들웨어 내부에 대한 상세한 구현에 대해서는 다루지 않고 다만 사용자 인터페이스를 기술함으로써 향후 상호 운용성 및 표준 적합성 검증을 용이하게 하였다. 본 논문에서는 ALE 스펙에 기반하여 RFID 데이터를 효율적으로 처리하고 서비스를 제공하기 위한 RFID 미들웨어를 설계한다.

본 논문의 구성은 다음과 같다. II장에서는 본 연구와 관련된 연구 및 기술에 대하여 살펴보고, III장에서는 RFID 미들웨어 설계에 대하여 살펴본다. 그리고 IV장에서는 결론과 향후 연구 방향에 대하여 설명한다.

## II. EPCglobal의 ALE 표준 스펙

### 1. ALE 개요

EPCglobal은 2003년 10월에 EAN과 UCC가 Auto-ID 센터(구 Auto-ID Lab)를 흡수하여 설립한 비영리 기구로서 Auto-ID에서 개발한 EPC 네트워크에 대한 기술의 표준화, 상용화, 그리고 EPC 코드 보급과 관리 등을 목적으로 활동하고 있다.

EPCglobal은 RFID 미들웨어의 표준화를 주도하고 있으며, 소프트웨어 관련 산업체에서는 이를 참조 모델로 솔루션을 개발하고 있다. EPCglobal은 RFID 태그로부터 외부 애플리케이션에 이르기까지의 구성 요소를 계층 구조로 표현하고 각 계층 간의 인터페이스를 표준화 대상으로 삼고 있다.

EPCglobal은 RFID 미들웨어와 관련하여 2002년에 구현 스펙 중심의 Savant Version 0.1[3]과 2003년에 Savant Version 1.0[2]를 제안하였고, 2005년 11월 다양한 센서로부터 EPC 코드 데이터를 받아 필터링하고 통합하여 클라이언트에게 제공하는 미들웨어의 인터페이스에 관한 ALE 스펙 버전 1.0을 발표하였다[4].

ALE 스펙은 미들웨어의 인터페이스(interface)에 관한 것이다. 또한 ALE 인터페이스는 응용과 물리 계층을 분리함으로써 기술 제공자와 사용자에게 비용 및 독립성 면에서 이점을 제공한다는 사실에 기반하여 설계되었다[5].

ALE 이전에는 데이터를 수집, 가공 처리하는 기능들은 Savant가 수행하였었다[3]. ALE는 Savant에서와는 달리 이 스펙에서는 구체적인 구현 전략 혹은 특정 소프트웨어 모듈 내에서의 내부 인터페이스를 기술하지 않는다. 대신 오로지 한 개의 외부 인터페이스만을 다룬다. 인터페이스를 따르기만 하면 어떻게 구현하든지는 미들웨어 개발자에게 맡긴다.

### 2. EPCglobal Network Architecture에서의 ALE 역할

ALE 계층은 데이터 수집하고 필터링하여 비즈니스 로직이 해석을 시작하도록 하는 의미있는 이벤트를 생성하는데 관심이 있다. 즉, 원시 EPC 데이터를 획득하는 하부 구조 모듈과 그 데이터를 필터링하고 카운팅하는 구조적 모듈, 그리고 데이터를 사용하는 클라이언트 응용 간의 독립성을 제공하는 것이다.

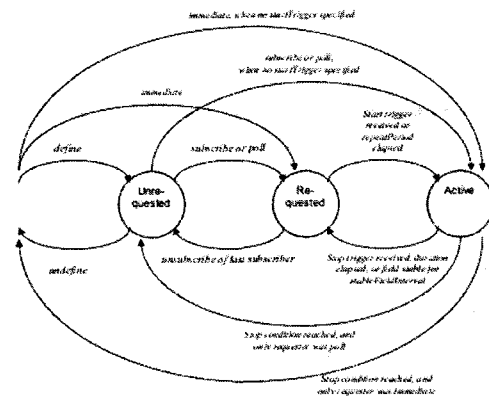
### 3. ALE API

#### 3.1 ALE API와 미들웨어 상태

일반적으로 하나 혹은 그 이상의 클라이언트들이 ALE 인터페이스에 있는 메소드를 호출한다. ALE 미들웨어는 그에 따른 적절한 동작을 수행하여 결과를 만들어 반환하는데, 여기에는 메소드 호출이 끝나야 결과를 반환하는 동기적인 특성을 갖는 Poll, Immediate 메소드가 정의되어 있다.

또한 비동기적으로 결과를 받을 수 있는 메소드로서 notificationURI를 파라미터로 받는 subscribe가 있다. 이 메소드는 호출과 동시에 바로 반환되며, 이후 미들웨어는 해당 연산을 수행한 후 그 결과를 URI에 명시되어 있는 클라이언트로 전달한다. 즉, 호출은 미리 끝나게 되고 미들웨어는 연산을 수행하는 대로 결과를 보내게 된다.

그림 4는 ALE 스펙에 정의되어 있는 여러 메소드 및 메소드 호출시 시스템의 상태가 어떻게 변하는지를 나타내는 상태 다이어그램이다.



▶▶ 그림 4. ALE 메소드 호출에 따른 상태 다이어그램

#### 3.2 이벤트 사이클

event cycle은 1개 혹은 그 이상의 리더에서 수행되는 1개 이상의 read cycle로 구성되는데, 이는 응용 관점에서 볼 경우 단일 동작으로 간주된다. 즉, event cycle은 ALE 인터페이스와 클라이언트 사이의 가장 작은 동작 단위이다. 참고로 클라이언트는 스펙에 event cycle 범위를 기술하여 ALE 미들웨어로 전송한다[4].

3.4 ALE 자료형

ALE API와 관련된 주요 자료 형으로 ECSpec과 ECRreport가 있다. ECSpec은 어떻게 event cycle이 수행되는지를 기술하며, ECRreport는 ECSpec을 활성화하여 생성한 하나 혹은 그 이상의 리포트들을 표현한다[4].

1) ECSpec

ECSpec은 하나의 event cycle과 그 event cycle로부터 어떤 리포트들이 생성되어야 하는지를 기술한다. 여기에는 어느 논리 리더로부터 EPC 데이터를 읽을 것인지에 대한 정보가 들어 있다. 즉, event cycle에 포함될 read cycle들을 수행하는 논리 Reader 리스트 정보가 기술되어 있다(readers). 또한 event cycle의 범위 정보(boundaries) 및 event cycle로부터 생성될 리포트에 대한 정보가 들어 있다[4].

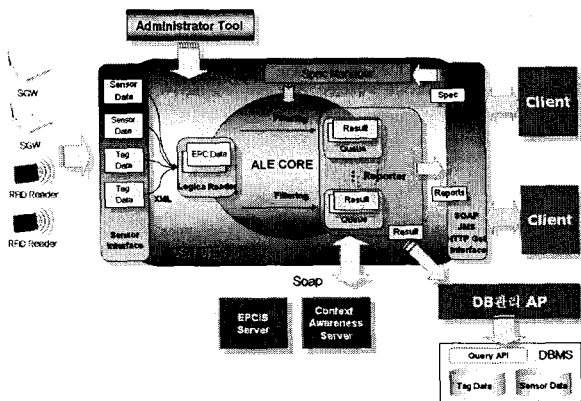
2) ECRreports

ECRreports는 한 개의 event cycle을 수행하여 생성한다[4]. ALE 미들웨어는 일정 기간 동안 데이터를 읽은 후 데이터를 필터링하고 카운트하여 리포트(들)를 반환한다. push 모드로 실행될 수도 있는데, 클라이언트는 어떤 데이터를 받는지에 대한 정보를 ECSpec을 작성하여 미들웨어에 등록하면 미들웨어는 event cycle이 완료될 경우 클라이언트로 리포트들을 비동기적으로 전송한다[4].

III. RFID 미들웨어 설계

1. 전체 구조도

그림 4는 본 논문에서 제안하는 RFID 미들웨어 시스템의 주요 요소 구조도이다. RFID 미들웨어 시스템은 크게 3부분으로 나누어 설명할 수 있다.



▶▶ 그림 10. 시스템 구성도

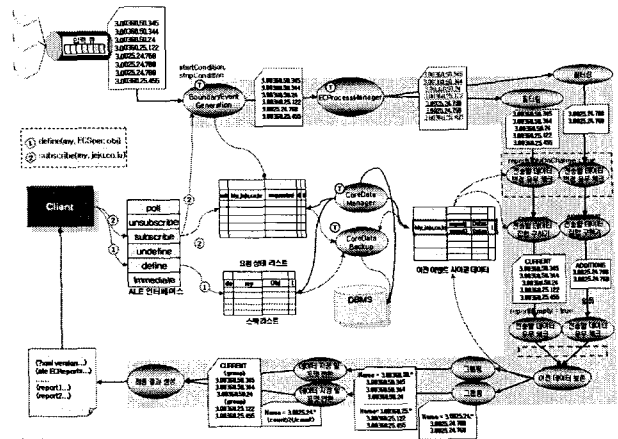
첫 번째 부분은 다양한 리더로부터 전달된 EPC 데이터를 정제하여 논리 리더로 생성하는 모듈, 두 번째 부분은 EPC 데이터를 응용에게 제공하는 ALE API, 세 번째 부분은 ALE 인터페이스를 통해 들어온 요청(ECSpec)을 내부적으로 필터링, 그룹핑 등 사용자의 요구에 맞게 연산, 정제하여 ECRreports로 생성하는 부분이다. 이러한 기능들은 ALE CORE 모듈이 수행한다.

2. 동작 시나리오 및 데이터 흐름도

그림 6은 EPC 데이터를 필요로 하는 사용자가 ALE 인터페이스를 이용하여 ECSpec을 정의하고 요청하였을 때 처리되는 데이터 흐름도이다. 동작 방식을 설명하면 다음과 같다. 우선 사용자는 ECSpec를 작성한 후 ALE 인터페이스의 define 메소드를 호출하여 스펙을 미들웨어 시스템에 등록한다.

define 메소드의 파라미터로 전송된 스펙은 스펙 리스트라는 자료 구조에 등록된다. 등록된 스펙 정보는 예상치 못한 시스템의 다운 및 이상이 발생할 경우 있을 수 있는 오류에 대비하기 위하여 일정한 시간 간격으로 백업 모듈에 의하여 데이터베이스에 백업된다.

이후 사용자는 미들웨어 시스템에 등록되어 있는 스펙을 이용하여 원하는 RFID 데이터를 요청할 수 있다. 이는 subscribe 메소드 호출을 통하여 이루어진다.



▶▶ 그림 6. 미들웨어 구성 모듈 및 데이터 흐름도

subscribe 메소드가 호출되면 스펙이 자료 구조에 등록되며, BoundaryEventGenerator 모듈은 스펙에 명시된 논리 리더에서 클라이언트가 필요로 하는 RFID 데이터를 꺼내어 ECProcessManager 모듈로 전송한다. BoundaryEventGenerator 모듈은 큐에 있는 태그 데이터를 가져올 때 중복된 데이터를 제거한다.

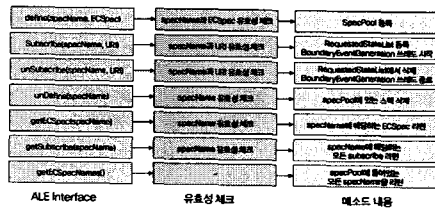
ECProcessManager 모듈은 넘겨받은 EPC 데이터에 대해

스펙에 명시된 필터링, 그룹핑, 데이터 셋과 전송할 유무체크 등 다양한 정제 과정을 수행하며, 지정된 데이터 포맷으로 변환한 후 최종 결과물인 ECReports 객체로 생성하여 데이터를 요청한 클라이언트 전송한다.

### 3. 주요 구성요소

#### 3.1 ALE Interface

ALE 인터페이스는 다음의 3가지를 통하여 하드웨어 독립성을 얻는다. 첫째, ALE 인터페이스는 클라이언트로 하여금 클라이언트가 관심을 갖는 EPC 데이터를 고 수준의 추상적인 방법으로 기술할 수 있도록 한다(ECSpec 작성). 둘째, 일반적으로 EPC 데이터는 어떤 하드웨어에서 생성되었고 어떻게 처리되었는지에 관계가 EPC 데이터를 표준 포맷으로 리포팅함으로써 이를 신경 쓰지 않도록 한다. 셋째, EPC 데이터 소스를 논리 리더기 개념 (논리 위치와 유사)으로 추상화시킴으로써 특정 논리 위치와 관련하여 어떤 하드웨어 장치를 사용하였는지 클라이언트가 신경을 쓰지 않도록 해준다. 그럼으로써 하드웨어가 바뀌더라도 클라이언트 응용에 영향이 미치지 않는다.



▶▶ 그림 12. ALE 인터페이스 처리 흐름도

#### 3.2 자료구조

##### 1) SpecStatesList 구조 - 2차원 동적배열

그림 8은 사용자가 등록한 ECSpec이 subscribe 메소드 호출시 스펙이 사용되어지고 있는 상태를 체크하고 관리하기 위한 자료구조이다.

Method Kind	specName	URI	specValue	State Kind	Dirty Kind
subscribe	mySpec	ubi.cheju.ac.kr	Requested	0	0

▶▶ 그림 13. SpecStatesList 구조 - 2차원 동적배열

##### 2) preDataList 구조 - HashMap

그림 9는 ECProcessManager에서 관리하고 있는 프로세스들이 사용하는 자료구조로 내부적으로 연산처리를 하기위한 임시저장 장소로 사용된다.

Method Kind	specName	URI	PreData	Dirty Kind
subscribe	mySpec	ubi.cheju.ac.kr	Null <Object>	0

▶▶ 그림 14. preDataList구조 - HashMap

#### 3.3 BoundaryEventGenerator

BoundaryEventGenerator는 사용자가 동기적 방법(immediate, poll) 혹은 비동기적 방법(subscribe)등 서비스 요청이 왔을 때 실행되며 ECSpec에 명시된 Boundary영역을 체크하여 사용자가 요청한 각종 조건을 내부처리에 필요한 변수에 값을 할당하는 역할과 시작과 종료 조건에 의해 스펙 상태 메모리에 Active & Requested 정보를 기록한다. 스펙 상태가 Requested -> Active 되면 Data Fetch를 실행하기 위한 상태 처리 비트를 0 -> 1, 메모리 상태를 DB에 저장을 위한 더티 비트를 0 -> 1 로 기록하고 스펙 상태가 Active -> Requested 되면 Data Fetch를 정지하기 위한 상태 처리 비트를 0 -> 1, 메모리 상태를 DB에 저장을 위한 더티 비트를 0 -> 1 로 기록한다. 또한 StartCondition 과 StopCondition을 체크하여 한 번의 이벤트사이클 동안 발생하는 EPC 데이터를 중복된 데이터를 제거 한 후 사용자의 요구에 맞는 데이터가공을 위해 ECProcessManager를 생성하고 데이터 덩어리를 넘기는 역할을 한다.

#### 3.4 ECProcessManager

BoundaryEventGenerator에 의해 생성되고 넘겨받은 한 번의 이벤트 사이클에서 얻은 데이터 덩어리를 사용자 요구에 맞게 연산처리를 하기위한 프로세스들을 관리 운영하는 역할을 한다.

#### 3.5 Backup Module

입력 값인 정보리스트를 가져와 데이터 백업 플래그를 체크하여 해당 항목을 데이터베이스에 저장, 업데이트 해주는 모듈

#### 3.6 필터링

중복이 제거된 데이터를 해당 스펙에 정의된 필터링 조건으로 데이터를 추출하는 기능

#### 3.7 ReportSet

스펙에 정의된 ReportSet 조건에 따라 현재 필터링 된 데이터와 이전 이벤트사이클에서 필터링 된 데이터를 비교하여 추가된 데이터 및 삭제된 데이터를 추출하는 기능

#### 3.8 ReportIfEmpty

ReportSet조건에 의해 추출된 데이터집합이(Send Buffer) 비어있어도 전송한다는 것을 체크한다.(reportIfEmpty = true) 디폴트 값은 false이다.(데이터가 없으면 보내지 않는 것이다.)

### 3.9 ReportOnChange

현재 ReportSet 조건에 의해서 추출된 데이터 집합이 이전 이벤트 사이클에서 추출된 데이터와 비교하여 변화된 것이 있는지를 체크하여 알려주는 기능(변경사항이 있을 경우 : true, 변경사항이 없을 경우 : false)

### 3.10 이전 데이터 보존

전송할 데이터를 리포트 이름을 키 값으로 preDataList에 일괄적으로 기록한다.

### 3.11 그룹핑

전송할 데이터를 그룹 패턴에 맞게 그룹으로 나누어 데이터로 썬다. 스펙 이름을 통해서, 그룹핑 조건을 알아내고 전송 데이터와 그룹핑 조건을 가지고 패턴 분석을 해서, 그룹 이름을 알아낸 후 그룹 이름을 통해서, 전송 데이터를 각 그룹 내임별 객체 배열에 저장하고 객체 배열들을 그룹버퍼에 저장한다.

### 3.12 데이터 지정 및 포맷 변환

스펙 이름을 통해서, outputSet 조건을 알아내고, outputSet 조건에 includeCount가 들어 있으면, 카운팅 작업을 한다. outputSet 조건에 groupList 조건이 들어 있으면, 그 조건에 해당하는 데이터 변환을 하고 변환된 데이터를 변환 버퍼에 저장한다.

### 3.13 DataManagerThread

입력값인 정보리스트를 가져와 데이터 백업 플래그와 메소드 플래그를 체크하여 해당 항목을 리스트에서 삭제 하는 모듈

설계하였다. 향후 계획으로는 본고에서 설계한 것을 기반으로 시스템을 구현하고 속도 및 비용절감에 필요한 관련연구가 필요할 것이다.

### ■ 참고 문헌 ■

- [1] EPCglobal, <http://www.epcglobalinc.org>
- [2] Auto-ID Labs, <http://www.autoidlabs.org>
- [3] Sean Clark and Ken Traub, etc., "Auto-ID Savant Specification Version 1.0", September 2003.
- [4] EPCglobal, "The Application Level Events (ALE) Specification Version 1.0", Ratified Specification Version of September 15, 2005.
- [5] Sun Microsystems, "The Sun EPC Network Architecture", Technical White Paper, February 2004.
- [6] 홍길동, "RFID 미들웨어를 위한 EMS 컴포넌트의 분석 및 설계", 한국정보과학회 한국컴퓨터종합학술대회 2005 논문집 (A), 2005.
- [7] Ken Trabub, Greg Allgair, Henri Barthel, Leo Burstenin.. "The EPCglobal Architecture Framework," EPCglobal, 2005.

## IV. 결론

본 논문에서는 먼저 ALE 스펙을 분석하여 EPCglobal Network Architecture 요소 간 인터페이스 정의에 중점을 두는 접근 방식이다. 기존 Savant는 미들웨어 내부에 대한 상세한 구현을 제시하여 RFID 미들웨어의 개발의 다양성과 용이성을 저해하는 요소가 되었고 그것을 이용하는 응용 개발자는 RFID 미들웨어에게 필요한 EPC 데이터 요청시 편의성을 보장 받지 못했다.

본 논문에서 설계하는 RFID 미들웨어는 ALE 표준을 준수하여 응용 개발자에게 ALE 인터페이스를 제공하여 동적으로 간편하게 요청스펙을 기술함으로써 미들웨어에 종속적이지 않고, 보다 효율적으로 RFID 시스템 운영이 가능하게 되었다. 즉 유연성과 확장성 있는 구조 설계로 RFID 시스템 간의 상호 운영성 및 표준 적합성 검증을 용이하게 하는 RFID 미들웨어를